

Application No.:	09/692,765	§	Examiner:	Patel, Ashokkumar B.
Filed:	October 19, 2000	§	Group/Art Unit:	2154
Inventors:		§	Atty. Dkt. No:	5181-65700/P5014
Thomas E. Saulpaugh, et al.		§		
		§		
		§		
		§		
Title:	Event Message Endpoints in	§		
	a Distributed Computing	§		
	Environment	§		
		§		
		§		
		§		
		§		

## **REPLY TO EXAMINER'S ANSWER**

### **First Ground of Rejection:**

Claims 1-11, 14-24, 27-33 and 36-46 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by Bass et al. (U.S. Patent 6,549,956) (hereinafter “Bass”). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

### **Claims 1, 11, 14, 24, 36 and 46:**

**Regarding claim 1, Appellant's have argued that, contrary to the Examiner's assertion, Bass fails to disclose all the limitations of Appellants' claims. For Example, Bass clearly fails to disclose receiving a message in a data representation language sent to a client platform in the distributed computing environment from a service in the distributed computing environment, wherein the message includes a data representation language representation of an event generated by the service.** Bass teaches a mechanism for connecting disparate publication and subscribe domains via the Internet in which two channel adapters act together as a bridge across the network. Specifically, Bass teaches the use of existing network protocols (e.g., SMTP, TCP/IP, etc) via existing holes in firewalls (column 2, lines 32-34). The Examiner cites portions of Bass (col. 2, lines 4-9, and 15-31, and column 3, lines 43-50) that describe how Bass' channel adapters receive published events, translate them into a message format suitable for transmission over a network, and send them to a channel adapter on another platform that translates the message back into the original event information.

The Examiner contends that by disclosing the translation of event information into network protocol messages, Bass discloses “that an event (message) can be represented in any data representation language and will be converted back into the event format for use in the other domain” (Office Action, page 3, lines 4-6). The Examiner further argues that Bass discloses that “an event (message) can be represented in any data

representation language”, referring to Bass’ teachings regarding channel adapters can convert event information into a format acceptable by the network. However, the Examiner reads too much into the actual teachings of Bass. The Examiner is arguing that the phrase “a format acceptable by the network” discloses the use of a data representation language. However, a data representation language is a particular type of language. As is well understood by anyone of ordinary skill in the art, a data representation language (such as XML) has a particular structure as a language, and is a language for representing (or describing) data or content. There is clearly no teaching in Bass that any messages are sent in a particular data representation language. Nor is there any teaching in Bass the events are represented in such a language. Typical interprocess communications do not involve a data representation language. Without some clear teaching by Bass regarding the use of a data representation language, Bass cannot be said to anticipate a message in a data representation language including a data representation language representation of an event. The Examiner’s hindsight-based speculation regarding the possible use of a data representation language in Bass for messages is clearly improper in a rejection based on anticipation under 35 U.S.C. § 102(e).

In the Response to Arguments, the Examiner refers to Bass’ teachings regarding sending event information in an email via SMTP. However, SMTP does not require the use of a data representation language. Data representation languages are well understood in the art. No one of ordinary skill in the art would consider SMTP (or any other similar network protocol) to be a data representation language. **The Examiner’s reference to the use of SMTP in Bass actually supports Appellants’ argument, since as is well understood by any one of ordinary skill in the art, the Simple Mail Transfer Protocol does not require any particular language, let alone the use of a data representation language.**

In the Examiner’s Answer, the Examiner argues, ““a message [in Bass] in a data representation language’ is the email, which is ‘the data representation language representation language of the event’” (Answer, page 21). Thus, the Examiner argues that the mere use of an email message somehow discloses the use of a data representation

language. However, Bass does not teach that email is communicated in a data representation language. Moreover, email messages are traditionally sent using plain text, which is not a data representation language. Electronic messaging certainly does not require the use of a data representation language. The mere fact that Bass describes sending email messages does not disclose or require that Bass' email message is in a data representation language. Nor does it disclose or require that the email message includes a data representation language representation of an event generated by the service.

Appellants have further argued that **Bass does not teach that the messages sent using these protocols are messages in a data representation language.** Instead, Bass teaches the translation of event information into existing network protocol messages, such as an SMTP email, TCP/IP packet, or FTP transfer message. Protocols and languages are not the same thing. Bass teaches the use of existing network protocols in order to take advantage of the fact that existing network protocols use existing holes in firewalls and other security mechanisms (see, column 2, lines 15-35). For instance, Bass teaches that an event is formatted for transmission on a network (such as the Internet) and that “[t]he format may use transmission control protocol/Internet protocol (TCP/IP), simple mail transport protocol (SMTP), File Transfer Protocol (FTP), or whatever protocol is useable by the connecting network” (column 3, lines 35-42). Thus, Bass is describing the use of *protocols*, not any particular *language*. Bass does not mention messages in a data representation language. Moreover, there is no reason to use messages in a data representation language in Bass's system, since none of the existing communications protocols advocated by Bass have any need for, or requirement, messages in a data representation language.

Data representation languages are *specific types of languages* traditionally used in the prior art to describe documents or other content (not to represent an event generated by a service in a distributed computing environment). XML is one example of a data representation language. Bass fails to mention anything about XML, **as admitted by the Examiner** in his statements regarding claim 13, or any other data representation language.

The cited art simply does not teach the use of a data representation language to represent events in messages between entities in a distributed computing environment. In the Examiner's Answer, the Examiner points out that Appellants' claim language makes no indication of XML. The Examiner has misunderstood Appellants' previous argument. Appellants are not arguing that claim 1 recites anything specifically about XML. Appellants' statement that Bass fails to mention XML supports Appellants' argument that Bass does not disclose anything about a data representation language, of which XML is a common example.

In response to the above argument, the Examiner has objected to Appellants' use of the phrase "data representation language message" to refer to a message in a data representation language, arguing that Appellants' claims do not recite the phrase "data representation language message." However, the Examiner's objection is clearly misplaced since Appellants' previous use of the phrase, "data representation language message" clearly refers to the "message in a data representation language" recited in Appellants' claim 1.

Additionally, Appellants' have argued that **Bass fails to anticipate that the message includes a data representation language representation of an event generated by the service**. In contrast, Bass teaches channel adapters that "convert the event information into a format acceptable by the network" (column 2, lines 15-18). The "format acceptable by the network" in Bass is not described as a data representation language representation of an event. Bass does not *disclose* a message that includes a data representation language representation of an event. The Examiner cites only col. 2, lines 4-9, and 15-31 of Bass that describe how a channel adapter translates event information into network protocol messages. Neither the Examiner's cited passages, nor any other portion of Bass, discloses, or even mentions, any data representation language representation of an event.

Furthermore, Appellants have argued that **Bass fails to anticipate sending the data representation language representation of the event to one or more processes registered to receive the event from the service**. The Examiner cites column 2, lines 9-15, where Bass describes a process adapter subscribing to and receiving an event via a channel adapter. However, Bass only teaches that the process adapter receives an event, not that it receives a data representation language representation of an event. The Examiner argues that by teaching a channel adapter that reformats an event for transmission over the Internet, Bass discloses the use of a *data representation language representation* of events. The Examiner's interpretation of Bass is incorrect. As noted above, Bass teaches only translating event information *into a format suitable for transmission over the Internet* via any of a number existing network *protocols* (such as TCP/IP, SMTP, FTP, etc). However, Bass is completely silent regarding a data representation language representation of an event.

In response the Appellants' above argument, the Examiner cites column 3, lines 43-50 of Bass describing how channel adapters receive published events, translate them into a message format suitable for transmission over a network, and send them to a channel adapter on another platform that translates the message back into the original event information. Thus, when Bass' channel adapter delivers the event to the subscribing process, the channel adapter has already, "re-transformed" the email (used to send the event information) back into the event (See, Bass, column 3, lines 45-50). Bass specifically states that the channel adapter delivers the re-constituted event, *rather than any **representation** of the event*, to subscribing processes, and certainly not a *data representation language representation of the event*. Thus, Bass fails to disclose sending the data representation language representation of the event to one or more processes registered to receive the event from the service.

Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The **identical invention** must be shown in as complete detail as is contained in

the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As shown above, Bass clearly fails to anticipate receiving a message in a data representation language sent to a client platform in the distributed computing environment from a service in the distributed computing environment, wherein the message includes a data representation language representation of an event generated by the service.

For at least the reasons given above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Arguments similar to those presented above regarding claim 1 apply to claims 14 and 36 as well.

**Claims 2, 7, 15, 21, 37 and 42:**

Regarding claim 2, Appellants have argued that **Bass fails to anticipate receiving a data representation language schema on the client platform, wherein the data representation language schema defines a message interface for a set of events generated by the service.** The Examiner cites column 3, lines 43-50 that describes Bass' channel adapters. Bass teaches that each channel adapter includes two interfaces, a framework interface and a protocol interface (column 3, lines 53-64). The framework interface includes domain specific protocols for communicating published and subscribed events with a domain broker. The protocol interface includes network specific protocols that enable the adapter to couple with the Internet. The Examiner has not cited any portion of Bass that teaches a *data representation language schema* defining a message interface for a set of events. Instead, Basses Bass teaches that each channel adapter includes two different interfaces for communicating event information, neither of which involves the client platform receiving a data representation language schema defining a message interface for a set of event, as recited in Appellants' claim.

The Examiner, in the Response to Arguments, also cites column 2, lines 4-15 and column 4, line 43 – column 5, line 15 where Bass teaches that each of his channel adapters are configured with a set of events it will export to a peer adapter in another domain. Bass teaches how a system administrator configures each channel adapter to

receive and transmit specific events and how channel adapters exchange, or export, lists of events that they will be communicating. The Examiner argues that exchanging event lists amounts to receiving a data representation language schema defining a message interface for a set of events. However, Bass' event list exchange only informs the channel adapter of *which events* will be communicated. Bass does not teach that his event export lists make up *a data representation language schema*. Bass also does not mention that the event export lists are exchanged using a data representation language. Furthermore, Bass does not describe his event export lists as defining message interfaces. To the contrary, as discussed above, Bass teaches (column 3, lines 53-64) that each channel adapter uses a protocol interface that includes network protocol messages. A channel adapter's protocol interface has nothing to do with the list of events that it may send and receive. Bass teaches that the channel adapter can convert any event into an appropriate network protocol messages. Thus, the exchange of exported event lists cited by the Examiner does not teach anything regarding receiving a data representation language schema defining a message interface.

Additionally, Appellants have argued that **Bass does not teach generating an event message endpoint for the client platform according to the data representation language schema**. The Examiner cites Bass' teachings regarding the receiving of events listed on an event type list (column 4, line 43 – column 5, line 15). The Examiner relies on Bass' teachings regarding an event on an event type list being received and re-published via the channel adapter and argues this discloses generating an event message endpoint according to a data representation language schema by describing how. Appellants have argued that the Examiner's interpretation of Bass is clearly incorrect. As discussed above, Bass' exported event type lists are not data representation language schemas. Moreover, not only do the event type lists in Bass fail to involve the generation of any message endpoints, they also have absolutely nothing to do with data representation language schemas. Thus, Bass clearly fails to disclose generating an event message endpoint for the client platform according to the data representation language schema.



In the Response to Arguments, the Examiner refers to the fact that in Bass, “an event is received via the channel adapter and re-published in to the domain” and that “the subscribing process adapter will receive the event.” The Examiner apparently equates Bass’ subscribing process adapter receiving an event with “generating an event message endpoint for the client platform according to the data representation language schema.” The Examiner’s argument seems to be that any adapter receiving an event must include an event message endpoint that was generated *according to a data representation language schema*. However, without some specific teaching by Bass regarding generating the process adapter *according to a data representation language schema* Bass cannot be said to disclose generating an event message endpoint for the client platform according to the data representation language schema, as recited by Appellants’ claim.

Thus, for at least the reasons give above, the Examiner’s rejection of claim 2 is not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 15 and 37 as well.

**Claims 3, 19 and 38:**

Regarding claim 3, Appellants have argued that **Bass fails to disclose the event message endpoint subscribing to one or more of the set of events generated by the service, wherein the service is configured to send messages including data representation language representations of an event to subscribers to the event when the event is generated.** The Examiner cites column 2, lines 4-15 of Bass describing how an event is delivered to a subscribing channel adapter. However, as discussed above regarding claims 1 and 2, Bass fails to teach anything regarding a service configured to send messages including *data representation language representations of events*. Instead, Bass teaches that events are converted into network protocol messages, such as SMTP email messages, for transmission over the Internet where they are converted back into the original event information for re-publishing in a different domain. Such network protocols are not data representation languages. Bass not only fails to describe these protocol messages as messages in a data representation language,

Bass also fails to mention a service configured to send messages including data representation language representations of an event.

In the Response to Arguments, the Examiner cites column 3, lines 49-50 where Bass states that channel adapters deliver events to any subscribing process adapters within the domain. However, as noted above, prior to delivering the event to the subscribing process adapters, the channel adapter converts the event information from the network protocol message (e.g. SMTP email message) back into the event. (see, Bass, column 3, lines 45-50). The event information delivered by the channel adapters is clearly not a data representation language representation of the event. Thus, the Examiner's cited passage actually supports Appellants' argument.

The Examiner also argues that Bass' process adapter can subscribe to an event type and that Bass' channel adapters are end points for receiving, as well as delivering, events. The Examiner concludes that Bass therefore discloses the limitations of Appellants' claim. The Examiner's argument relies on Bass' process adapter subscribing to events *from the channel adaptor*. However, as noted above, Bass' channel adapters convert event information from the network protocol message (e.g. SMTP email message) back into the event before delivering the event to a subscribing process. Thus, Bass' process adapter cannot be considered an event message endpoint subscribing to one or more of the set of events generated by a service, wherein the service is configured to send messages including data representation language representations of an event to subscribers to the event when the event is generated. Furthermore, the Examiner has failed to acknowledge or address this argument.

For at least the reasons above, the rejection of claim 3 is not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 19 and 38 as well.

**Claims 4, 20 and 39:**

In regards to claim 4, Appellants have argued that **Bass fails to disclose wherein the data representation language message from the service includes an authentication credential for the service. Bass additionally fails to disclose the event message endpoint using the authentication credential for the service to authenticate the data representation language message as being from the service.** The Examiner cites column 4, line 57 to column 5, line 15 of Bass that describes how Bass' channel adapters are configured to send and receive various events. Please see the discussion above regarding claim 2 for a more detailed discussion of this portion of Bass. The Examiner does not provide any argument or discussion regarding how Bass' exported event type lists have relevance to a data representation language message including an authentication credential. Nowhere does Bass mention anything regarding data representation language messages including authentication credentials nor about event message endpoints using an authentication credential to authenticate the data representation language message.

In the Response to Arguments, the Examiner cites column 1, lines 56-60 of Bass and argues that Bass' stated purpose for his invention, namely, "there is a need in the art for a mechanism to link to disparate PUB/SUB domains together without compromising security, reducing performance, be easy to implement, and still allow for information transfer between the two domains" discloses the specific limitations of Appellants' claim 4. However, a general statement regarding Bass intention that his invention no compromise security does not disclose or anticipate the specific limitation of a data representation language message from a service including an authentication credential for the service. Nor does the cited passage anticipate an event message endpoint using the authentication credential for the service to authenticate the data representation language message as being from the service.

In the Response to Arguments, the Examiner further asserts that each of Bass' channel adapters includes a configuration interface allowing the channel adapter to be

configured to an administrator, citing column 3, line 64 through column 4, line 3. Specifically, the Examiner argues that the channel adapter's configuration interface discloses "service includes an authentication credential for the service." Apparently, the Examiner's argument is that an administrator being able to select the protocol used by Bass' configuration interfaces somehow *discloses* the use of an authentication credential. Firstly, Appellants' claim does not recite that a "service includes an authentication credential for the service". Instead, Appellants' claim recites that the "data representation language message from the service includes an authentication credential for the service". Secondly, the mere fact that Bass' channel adapter includes a configuration interface does not disclose anything about Bass' an event message (which the Examiner erroneously equates to a data representation language message) including an authentication credential for the service, which would be required for the Examiner's interpretation to be correct. Thirdly, the fact that Bass' channel adapter includes a configuration interface does not have any relevance to the fact that Bass fails to disclose the event message endpoint using the authentication credential for the service to authenticate the data representation language message as being from the service.

As noted above regarding claim 1, anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. The passages cited by the Examiner can in no way be considered to disclose each and every element of Appellants' claim 4, arranged as in the claim. Thus, for at least the reasons above, the rejection of claim 4 is not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 20 and 39 as well.

**Claims 5, 16 and 40:**

Regarding claim 5, Appellants have argued that **Bass fails to disclose the event message endpoint verifying type correctness of the data representation language message according to the data representation language schema**. The Examiner cites column 2, lines 24-27 and column 3, lines 45-50 of Bass. However, the first cited portion

only describes how Bass' channel adapters use a plurality of states and status messages to track and indicate the delivery, receipt, and publication of events. The second cited portion describes how an event is transformed into an email message via SMTP and then re-transformed back into the event upon receipt. Thus, neither of the Examiner's cited portions has anything to do with *verifying type correctness* of a data representation language message *according to a data representation language schema*. The various states and status messages indicating delivery, receipt, and publication of events only track and help to guarantee that event messages are eventually delivered to the subscribing process adapter. These states have nothing to do with verifying type correctness of a data representation language message according to a data representation language schema. Similarly, converting an event into an email message via SMTP (or another network protocol message) and converting the message back into an event has nothing to do with verifying type correctness of a data representation language message according to a data representation language schema. In fact, nowhere does Bass make any reference whatsoever to verifying type correctness of a data representation language message according to a data representation language schema.

Furthermore, the Examiner has argued that Bass' exported event type lists constitute a data representation language schema (see, Final Office Action, pages 14-15, regarding claim 2). However, Bass does not teach that an exported event type list has anything do with the states and status messages indicating delivery, receipt, and publication of events or have anything to do with converting events into SMTP messages. Thus, the Examiner's interpretation of Bass is inconsistent and thus cannot be correct.

In the Response to Arguments, the Examiner cites column 4, lines 18-24 where Bass teaches that each channel adapter includes a reporting mechanism. Bass describes this reporting mechanism as informing an administrator of the status of events and that the administrator can "determine if there are any events that are stuck, and the state in which they are stuck." However, reporting on the state of events as they flow through Bass' system does not disclose the specific functionality of *verifying type correctness of a data representation language message according to a data representation language*

*schema*, as recited in Appellants' claim. Not only does the cited passage fail to mention verifying type correctness of any messages, the passage also fails to mention anything regarding the use of a data representation language schema to verify type correctness.

In the Response to Arguments, the Examiner refers to Bass' teaching regarding channel adapters including a guaranteed message delivery mechanism that receives stores and transmits events as well as monitors and tracks delivery states of events, citing column 4, lines 4 – 17. However, guaranteeing that a message is delivered, including tracking delivery states does not have anything to do with verifying type correctness of a data representation language message according to a data representation language schema. Guaranteeing that a message is delivered merely ensures that the message is received, but does not verify any particular type correctness according to a schema. An improperly typed message may still be delivered.

The Examiner also responds to the above argument by repeating the same response from the Response to Argument section of the Final Office Action and the same citation from Bass (column 4, lines 18-24). However, as noted above, this passage of Bass has no relevance to verifying the type correctness of messages using a data representation language schema. The Examiner has not provided any additional arguments regarding how Bass' discussion of stuck messages involves verifying type correctness. Nor has the Examiner cited any additional portion of Bass that mentions using a data representation language schema to verify type correctness.

Additionally, the reporting mechanism relied on by the Examiner is not Bass' Channel Adapter, which the Examiner has previously equated to the event message endpoint of Appellants' claims (see Examiner's rejection of claim 2, Final Office Action, page 14, lines 4-10). Appellants' claim 5 recites the event message endpoint verifying type correctness of the message according to a data representation language schema. Even if Bass' Channel Adapter were to verify type correctness of message according to a data representation language schema, which it does not, Bass still fails to anticipate Appellants' claim 5.

The Examiner's has clearly failed to provide a proper rejection based on anticipation. The rejection of claim 5 is not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 16 and 40 as well.

**Claims 6, 17 and 41:**

Regarding claim 6, Appellants have argued that Appellants have argued that **Bass fails to anticipate wherein the data representation language schema defines a set of messages that the service may send to the event message endpoint and further fails to teach the event message endpoint verifying the correctness of the data representation language message from the service according to the data representation language schema**. The Examiner once again cites column 2, lines 24-27 and column 3, lines 45-50 of Bass. However, as noted above regarding claim 5, the first cited portion only describes how Bass' channel adapters use a plurality of states and status messages to indicate the delivery, receipt, and publication of events and the second cited portion describes how an event is transformed into an email message via SMTP and then re-transformed back into the event upon receipt. As discussed above regarding claim 5 (for which the Examiner cites the same portions of Bass), neither of the Examiner's cited portions have anything to do with a data representation language schema defining a set of messages that a service may send to an event message endpoint. Additionally, neither of the cited passages mentions an event message endpoint verifying the correctness of a data representation language message according to the data representation language schema.

Specifically, the various states and status messages (indicating delivery, receipt, and publication of events) only help to track and guarantee that event messages are eventually received by the subscribing process adapter. These states have nothing to do with verifying the correctness of a data representation language message according to a data representation language schema. Similarly, converting an event into an email

message via SMTP (or another network protocol message) and converting the message back into an event is not verifying type correctness of a data representation language message according to a data representation language schema.

The Examiner argues, in the Response to Arguments, that Bass' channel adapters include protocol interfaces facilitation conversion of events into a network transportable format discloses the limitations of Appellants' claim. However, merely because Bass' channel adapters convert events into network transportable formats, such as SMTP, does not disclose anything regarding a data representation language schema that defines a set of messages, as recited in Appellants' claims. Events may be converted to network transportable format without the use of a data representation language schema. Furthermore, the conversion of events to network transportable formats does not disclose anything about an event message endpoint verifying the correctness of a data representation language message from a service according to a data representation language schema, as recited by Appellants' claims. The Examiner has not presented any argument or interpretation of Bass that discloses anything about verifying the correctness of a data representation language message according to a data representation language schema.

Thus, the rejection of claim 6 is clearly not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 17 and 41 as well.

**Claims 8, 22 and 43:**

Regarding claim 8, Appellants have argued that **Bass fails to disclose each of the one or more processes providing an event handler callback method to the event message endpoint**. The Examiner cites column 4, lines 57-60. However, this portion of Bass only teaches that Bass' channel adapters subscribe to and publish events, but fails to describe any mechanism for delivering the events other than via existing network



protocols. Nowhere does Bass teach providing an event handler callback method to an event message endpoint.

In the Response to Arguments, the Examiner cites column 4, line 43 through column 5, line 16 and refers to Bass' teaching that channel adapters republish events to interested process adapters. Specifically the Examiner refers to Bass' teaching that "prior to transfer of events between the domains, the respective process and channel adapters of the domains must be configured to send and receive the different events" (Bass, column 4, lines 57-59). However, merely stating that the process and channel adapters must be configured to send and receive events in no way discloses, teaches, or even implies providing an event handler callback method to an event message endpoint. Without some clear and specific teaching by Bass regarding providing an event handler callback method, the Examiner is merely speculating, which is clearly improper, as to the details of Bass system.

Appellants have also argued that **Bass further fails to teach the event message endpoint calling an event handler method of each process registered with the event message endpoint and the event message endpoint passing the data representation language representation of the event to each called event handler.** The Examiner cites column 3, lines 22-50 where Bass describes how his channel adapters convert events to and from network protocol message and how events are sent over the Internet and republished in other domains. However, nowhere does Bass describe an event message endpoint calling an event handler method. Nor does Bass teach an event message endpoint passing a data representation language representation of an event to each called event handler. The Examiner merely states, "the reference teaches that the processes as well as the adapters are configured to do the claimed element" and further contends, "the c[h]annel adapters are capable of executing the task as claimed." However, without any supporting teaching from Bass, the Examiner's rejection amounts to nothing more than mere hindsight speculation and conclusory statements.

In the Response to Arguments, the Examiner refers to the fact that, “prior to transfer of events between the domains, the respective process and channel adapters of the domains must be configured to send and receive the different events.” The Examiner then asserts, “[t]hus, ‘the event message endpoint [calling] an event handler method of each process registered with the event message endpoint and the event message endpoint passing the data representation language representation of the event to each called event handler’ is understood”. However, the Examiner fails to show why Bass’ process and channel adapters being configured to send and receive the different events necessarily includes an event message endpoint calling an event handler method of each process registered with the event message endpoint to the event, as would be required for Bass to anticipate Appellants’ claim. The Examiner’s conclusory statements do not provide a proper *prima facie* anticipation rejection.

Thus, for at least the reasons given above, the rejection of claim 8 is not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 22 and 43 as well.

**Claims 9, 23 and 44:**

Regarding claim 9, Appellants have argued that **Bass fails to disclose, contrary to the Examiner’s assertion, a process unregistering interest in a first event of the service.** The Examiner cites column 4, line 57 through column 5, line 15 of Bass. However, this passage of Bass only refers to delivering events to channel adapters that have subscribed to receive the event. The passage cited by the Examiner does not mention anything about a process unregistering interest in an event. In fact, nowhere does Bass mention a channel adapter, or other process, unsubscribing to events.

Additionally, Appellants have argued that **Bass fails to the event message gate unsubscribing to the event with the service subsequent to the unregistering.** The Examiner fails to cite any portion of Bass describing an event message gate unsubscribing to an event. The Examiner’s cited passage, described above, fails to

mention anything about an event message gate, or any other component of Bass' system, unsubscribing to events. Instead, as noted above, the cited passage only refers to delivering events to subscribing channel adapters. As noted above, Bass fails to teach anything about unsubscribing to events.

In the Examiner's Answer, the Examiner again cites Bass, column 4, line 57 through column 5, line 15. The Examiner describes how an administrator in Bass' system configures a broker to list a channel adapter as a subscriber to an event and how the channel adapter would then send an updated export list to other channel adapters. The Examiner then describes that the administrator would also configure the channel adapter to publish the event. The Examiner then merely states, "[t]hus the event message gate unsubscribing as well as subscribing to the event with the service subsequent to the unregistering is taught by Bass." However, none of the Examiner's discussion of the cited passage makes any reference whatsoever regarding a process unregistering interest in an event or about an event message gate unsubscribing to the event. The Examiner has merely stated, without any supporting teaching by Bass, that Bass teaches the limitations of Appellant's claim, which is clearly improper in a rejection based on anticipation.

Thus, the rejection of claim 9 is not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 23 and 44 as well.

**Claims 10, 18 and 45:**

Regarding claim 10, Appellants have argued that **Bass does not disclose receiving the data representation language schema of the service in a service advertisement of the service**. The Examiner cites column 2, lines 4-15, column 3, lines 43-50 and column 4, line 43 – column 5, line 15, where Bass describes how his channel adapters are configured with a set of events it will export to its peer adapter in another domain. Please refer to the remarks above regarding claim 2 for a discussion of these portions of Bass. The Examiner apparently contends that Bass's use of exported event

type lists include service advertisements. However, the exported event type lists have absolutely nothing to do with a service advertisement that includes a data representation language schema.

In the Response to Arguments, the Examiner cites column 2, lines 4-15, column 3, lines 43-50 and column 4, line 43 – column 5, line 15 of Bass, without providing any additional argument or interpretation regarding Appellants' argument above.

Also in the Response to Arguments, the Examiner describes Bass' channel and process adapters, Bass' mechanism for configuring channel and process adapters, and Bass method for delivering events. However, nowhere does the Examiner cite any passage of Bass or provide any interpretation of Bass to support the erroneous contention that Bass discloses receiving a data representation language schema of the service in a service advertisement of the service. Bass makes no mention of any service advertisement. Bass does not describe receiving a data representation language schema of a service in a service advertisement. The Examiner's hindsight-based speculation regarding receiving a data representation language schema of the service in a service advertisement of the service in Bass is clearly improper in a rejection based on anticipation under 35 U.S.C. § 102(e).

For at least the reasons given above, the rejection of claim 10 is not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 18 and 45 as well.

**Claims 27 and 28:**

Regarding claim 27, Appellants have argued that **Bass fails to anticipate a service process configured to generate a message in a data representation language**. The Examiner cites column 2, lines 49, and 15-31 of Bass and argues that converting event information into a format acceptable by the network discloses that an event (message) can be represented in any data representation language. The Examiner's

interpretation of Bass is incorrect. As discussed above regarding claim 1, Bass teaches the use of existing network protocols such as SMTP, TCP/IP, or FTP which have absolutely no bearing whatsoever on the use of a data representation language. Bass does not mention anything regarding using data representation language messages.

Additionally, Appellants have also argued that **Bass fails to anticipate wherein the message includes a data representation language representation of the event generated by the service process**. The Examiner does not cite any passage in Bass that refers to a message including a data representation language representation of an event, as suggested by the Examiner. Instead, Bass teaches that the event information is translated into a network protocol message, as described above regarding claim 1. Furthermore, as defined in the art, existing network protocols do not include data representation language representation of events.

Appellants have also argued that **Bass does not anticipate wherein each of the one or more event message gate units is operable to distribute the data representation language representation of the event**, as asserted by the Examiner. Also as noted above regarding claim 1, Bass teaches that once received by a channel adapter, the network protocol message is converted back into the original event information. Thus, in order to distribute data representation language representations of an event, an event would have to originally be a data representation language representation of the event. However, Bass does not teach anything regarding data representation language representations of events. The Examiner has not cited any passage of Bass that refers to data representation language representations of an event.

In the Response to Arguments, the Examiner cites elements 16 and 17 of Bass' FIG. 1 and refers to the previous Response to Arguments regarding claim 1. However, FIG. 1 of Bass does not illustrate a message in a data representation language or data representation language representations of events generated by the service process. Furthermore, as noted above, Bass teaches the use of existing network protocols, such as SMTP, TCP/IP, or FTP, which, as discussed above, are not data representation languages.

In the Examiner's Answer, the Examiner repeats the same response and arguments from the Examiner's response to Appellants' arguments regarding claim 1, discussed above. Please refer to Appellants' discussion of claim 1 above regarding Examiner's remarks from the Examiner's Answer.

For at least the reasons given above, the rejection of claim 27 is not supported by the prior art and removal thereof is respectfully requested.

**Claim 29:**

Regarding claim 29, Appellants have argued that **Bass fails to anticipate a service process configured to provide a data representation language schema defining a message interface for a set of events generated by the service and also fails to teach wherein one or more event message gate units are generated according to the data representation language schema**. The Examiner cites column 3, lines 43-50 that describes Bass' channel adapters. Bass teaches that each channel adapter includes two interfaces, a framework interface and a protocol interface (column 3, lines 53-64). The framework interface includes domain specific protocols for communicating published and subscribed events with a domain broker. The protocol interface includes network specific protocols that enable the adapter to couple with the Internet. The Examiner has not cited any portion of Bass that teaches a data representation language schema defining a message interface for a set of events. Instead, Bass teaches that each channel adapter includes two different interfaces for communicating event information.

The Examiner also cites and column 2, lines 4-15 and column 4, line 43 – column 5, line 15 where Bass teaches how his channel adapters are configured with a set of events it will export to its peer adapter in another domain. Bass teaches how an administrator configured each channel adapter to receive and transmit specific events and how channel adapters exchange lists of events that they will be communicating. The Examiner argues that exchanging event lists amounts to receiving a data representation

language schema defining a message interface for a set of events. However, Bass' event list exchange only informs the channel adapter which events will be communicated. Bass does not teach that his event export lists are data representation language schemas. Bass does not mention that the event export lists are exchanged using a data representation language. Furthermore, Bass does not describe his event export lists as defining a message interfaces. On the contrary, as discussed above, Bass teaches (column 3, lines 53-64) that each channel adapter includes a protocol interface that includes network protocol messages. A channel adapter's protocol interface has nothing to do with the list of events that it may send and receive. Bass teaches that the channel adapter can convert any event into an appropriate network protocol messages. Thus, the exchange of exported event lists cited by the Examiner does not teach anything regarding receiving a data representation language schema defining a message interface.

Appellants have further argued that **Bass also fails to teach generating event message gate units according to a data representation language schema**. The Examiner cites Bass' teachings regarding the receiving of events listed on an event type list (column 4, line 43 – column 5, line 15) and argues that Bass discloses generating event message gate units according to a data representation language schema by describing how an event on an event type list is received and re-published via the channel adapter. The Examiner's interpretation of Bass is clearly incorrect. As discussed above, Bass' exported event type lists are not data representation language schemas. Not only do the event type lists fail to define any message interfaces, they also do not use a data representation language.

In the Examiner's Answer, the Examiner repeats the same response and arguments from the Examiner's response to Appellants' arguments regarding claim 5, discussed above. Please refer to Appellants' discussion of claim 5 above regarding Examiner's remarks from the Examiner's Answer.

Thus, for at least the reasons given above, the Examiner's rejection of claim 29 is not supported by the prior art and removal thereof is respectfully requested.

**Claim 30:**

Regarding claim 30, Appellants have argued that **Bass fails to anticipate wherein the data representation language schema defines a set of messages that the service may send to the event message gate units.** The Examiner cites column 2, lines 24-27 and column 3, lines 45-50 of Bass. However, as noted above regarding claims 5 and 6, the first cited portion only describes how Bass' channel adapters use a plurality of states and status messages to indicate the delivery, receipt, and publication of events and the second cited portion describes how an event is transformed into an email message via SMTP and then re-transformed back into the event upon receipt. As discussed above regarding claim 5 (for which the Examiner cites the same portions of Bass), neither of the Examiner's cited portions have anything to do with a data representation language schema defining a set of messages that a service may send to an event message gate units. Bass is silent regarding a data representation language schema defining a set of messages that a service may send to the event message gate units.

In the Examiner's Answer, the Examiner refers to the Bass' teaching regarding an administrator configuring a broker to list a channel adapter as a subscriber to an event, citing column 4, lines 64 – 66. However, an administrator configuring a broker to list a channel adapter as a subscriber to an event does not disclose anything regarding a data representation language schema defining a set of messages that a service may send to event message gate units.

Thus, the rejection of claim 30 is clearly not supported by the prior art and removal thereof is respectfully requested.

**Claim 31:**

Regarding claim 31, Appellants have argued that **Bass does not teach a service process configured to provide the data representation language schema in a service**



**advertisement.** The Examiner cites column 2, lines 4-15, column 3, lines 43-50 and column 4, line 43 – column 5, line 15, where Bass describes how his channel adapters are configured with a set of events it will export to its peer adapter in another domain. Please refer to the remarks above regarding claim 2 for a discussion of these portions of Bass. The Examiner apparently contends that Bass use of exported event type lists include service advertisements. However, the exported event type lists have absolutely nothing to do with a service advertisement that includes a data representation language schema.

In the Examiner's Answer, the Examiner describes Bass' channel and process adapters, Bass' mechanism for configuring channel and process adapters, and Bass method for delivering events, citing column 2, lines 4-15, column 3, lines 43-50 and column 4, line 43 – column 5, line 15 of Bass. However, nowhere does the Examiner cite any passage of Bass or provide any interpretation of Bass to support the erroneous contention that Bass discloses providing a data representation language schema of the service in a service advertisement. Bass makes no mention of any service advertisement. Bass does not describe providing a data representation language schema of a service in a service advertisement. The Examiner's hindsight-based speculation regarding receiving a data representation language schema of the service in a service advertisement of the service in Bass is clearly improper in a rejection based on anticipation under 35 U.S.C. § 102(e).

Thus, for at least the reasons given above, the rejection of claim 31 is not supported by the prior art and removal thereof is respectfully requested.

**Claim 32:**

Regarding claim 32, Appellants have argued that **Bass fails to teach the event message endpoint subscribing to one or more of the set of events generated by the service, wherein the service is configured to send messages including data representation language representations of an event to subscribers to the event when the event is generated.** The Examiner cites column 3, lines 43-50 of Bass

describing how an event is delivered to a subscribing channel adapter. However, as discussed above, Bass fails to teach anything regarding a service configured to send message including data representation language representations of an event. Instead, Bass teaches that events are converted into network protocol messages for transmission over the internet where they are converted back into the original event information for re-publishing in a different domain. Nowhere does Bass mention a service sending messages including data representation language representations of an event.

In the Examiner's Answer, the Examiner repeats the same response and arguments from the Examiner's response to Appellants' arguments regarding claim 3, discussed above. Please refer to Appellants' discussion of claim 3 above regarding Examiner's remarks from the Examiner's Answer.

Thus, the rejection of claim 32 is not supported by the prior art and removal thereof is respectfully requested.

**Claim 33:**

In regards to claim 33, Appellants have argued that **Bass fails to disclose wherein the service process is further configured to attach an authentication credential for the service to the data representation language message, where the authentication credential is configured for use in authenticating the data representation language message as being from the service process**. The Examiner cites column 4, line 57 to column 5, line 15 of Bass that describes how Bass' channel adapters are configured to send and receive various events. Please see the discussion above regarding claim 2 for a more detailed discussion of this portion of Bass. The Examiner does not provide any argument or discussion regarding how Bass' exported event type lists have relevance to a data representation language message including an authentication credential. Nowhere does Bass mention anything regarding data representation language messages including authentication credentials nor about event message endpoints using an authentication credential to authenticate the data representation language message.

In the Response to Arguments, the Examiner cites column 1, lines 56-60 of Bass and argues that Bass' stated purpose for his invention, namely, "there is a need in the art for a mechanism to link to disparate PUB/SUB domains together without compromising security, reducing performance, be easy to implement, and still allow for information transfer between the two domains" discloses the specific limitations of Appellants' claim 4. However, a general statement regarding Bass' intention that his invention no compromise security does not disclose or anticipate the specific limitation of a data representation language message from a service including an authentication credential for the service. Nor does the cited passage anticipate an event message endpoint using the authentication credential for the service to authenticate the data representation language message as being from the service.

In the Examiner's Answer, the Examiner asserts that each of Bass' channel adapters includes a configuration interface allowing the channel adapter to be configured to an administrator. The Examiner argues that the channel adapter's configuration interface somehow discloses "service includes an authentication credential for the service." Firstly, Appellants' claim does not recite that a "service includes an authentication credential for the service". Instead, Appellants' claim recites that the "service process is further configured to attach an authentication credential for the service to the data representation language message, where the authentication credential is configured for use in authenticating the data representation language message as being from the service process". Secondly, the mere fact that Bass' channel adapter includes a configuration interface does not disclose anything about Bass' event messages, which the Examiner erroneously equates to a data representation language message, including an authentication credential for the service, which would be required for the Examiner's interpretation to be correct. Thirdly, the fact that Bass' channel adapter includes a configuration interface does not have any relevance to the fact that Bass fails to disclose a service process configured to attach an authentication credential for the service to the data representation language message, where the authentication credential is configured

for use in authenticating the data representation language message as being from the service process.

As noted above, anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. The passages cited by the Examiner can in no way be considered to disclose each and every element of Appellants' claim 33, arranged as in the claim. Thus, for at least the reasons above, the rejection of claim 33 is not supported by the prior art and removal thereof is respectfully requested.

### **Second Ground of Rejection:**

Claims 12, 13, 25, 26, 34, 35, 47 and 48 stand finally rejected under 35 U.S.C. § 103(a) as being anticipated by Bass in view of Meltzer et al. (U.S. Patent 6,542,912) (hereinafter "Meltzer"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

#### **Claim 12, 25 and 47:**

Appellants have argued that, contrary to the Examiner's assertion, **Bass in view of Meltzer fails to teach or suggest a message in a data representation language including a data representation language representation of an event, where the event is a JAVA event**. The Examiner admits that Bass does not disclose the use of XML or JAVA events. The Examiner relies upon Meltzer, citing column 14, lines 25-32. Meltzer teaches a system for exchanging business related information using self-defining documents, such as XML based documents. The Examiner's cited passage describes turning an XML document into a set of Java events. However, Meltzer teaches that listeners in a publish and subscribe architecture do not have to understand XML, but instead may be JAVA based and Meltzer specifically teaches the translation of XML documents into JAVA events to allow listeners to process received documents (Meltzer, column 14, lines 41-54). Thus, rather than teaching or suggesting using data

representation language representations of JAVA events, as the Examiner contends, **Meltzer teaches the opposite**, using Java events to represent a data representation language (e.g. XML) document. **The Examiner failed to rebut this argument.**

Thus, the combination of Bass and Meltzer would not result in a publish and subscribe system in which messages including data representation language representations of JAVA events, as recited in Appellants' claims. Instead, the Examiner's proposed combination of Bass and Meltzer would result in a publish and subscribe system in which data representation language documents are translated into Java events for listeners or other recipients of the documents.

Furthermore, since Bass fails to teach the use of XML or JAVA events, as the Examiner admits, it would not make sense, nor would it be obvious, to modify the system of Bass to translate XML documents into JAVA events as taught by Meltzer. The Examiner's stated motivation, namely, so that a transaction process front end is able to operate in a publish and subscribe architecture that enables the addition of new listener programs without the knowledge of or impact on other listening programs in the system" is merely a reason why someone would use Meltzer's system and does not provide any reason to modify the teachings of Bass. As noted above, the Examiner is relying upon the teachings of Meltzer regarding translating XML documents to JAVA events. However, since Bass does not teach anything about either XML documents or JAVA events, there is no reason to modify Bass' system to include these teachings of Meltzer.

In the Examiner's Answer, the Examiner completely ignores Appellants' arguments regarding Meltzer teaching the opposite of what the Examiner is relying up Meltzer to teach. The Examiner also ignores Appellants' arguments regarding a lack of a reason to modify Bass' system to include the teachings of Meltzer. Instead, the Examiner again refers to Meltzer's teachings regarding "translating the event in a data representation language into a JAVA event." The Examiner has failed to address the fact that Appellants' claim 12 recites (in conjunction with claim 1, from which claim 12 depends), in part, "receiving a message in a data representation language ... wherein the

message includes a data representation language representation of an event ... wherein the event is a Java event.” As noted above, the Examiner’s cited art fails to teach or suggest a message in a data representation language that includes a data representation language representation of a JAVA event. The Examiner’s argument relies on Meltzer’s teachings regarding translating XML documents into JAVA events so that do not understand XML, a data representation language, may process the received documents in their JAVA event form.

Therefore, the rejection of claim 12 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks also apply to claims 25 and 47.

**Claim 13, 26 and 48:**

Regarding claim 13, Appellants have argued that the Examiner’s combination of **Bass in view of Meltzer fails to teach or suggest receiving a message in a data representation language sent to a client platform in the distributed computing environment from a service in the distributed computing environment, wherein the message includes a data representation language representation of an event generated by the service, wherein the data representation language is eXtensible Markup Language (XML)**. The Examiner admits that Bass fails to teach using XML and relies upon Meltzer, citing column 14, lines 25-32. As noted above regarding the rejection of claim 12, Meltzer teaches a system for exchanging business related information using self-defining documents, such as XML based documents. The Examiner’s cited passage describes turning an XML document into a set of Java events. Thus, Meltzer does not teach or suggest sending messages in a XML, nor does Meltzer teach or suggest including an XML representation of an event in such a message. In fact, Meltzer teach away from including XML representations of events. Instead, as described above regarding claim 12, Meltzer teaches translating XML documents into JAVA events. Nowhere does Meltzer teach or suggest including a data representation language representation of an event in a message in a data representation language, where the data representation language is XML.

Since neither Bass nor Meltzer teach or suggest using XML for message or including an XML representation of an event in a message, the Examiner's combination of Bass and Meltzer also fails to teach or suggest sending messages in a XML, nor does Meltzer teach or suggest including an XML representation of an event in such a message. Instead, the Examiner's combination of Bass and Meltzer would result in the publish and subscribe system taught by Bass that also includes translating XML documents into JAVA events, as taught by Meltzer.

In the Examiner's Answer, the Examiner again cites column 14, lines 25-32 of Meltzer and refers to Meltzer's simple statement that XML-based documents may be easily understood amongst trading partners. However, the Examiner has failed to respond to Appellants' arguments regarding the fact that neither Bass nor Meltzer, whether considered singly or in combination, teach or suggest using XML for message or including an XML representation of an event in a message. Meltzer's statement that XML documents are used to "tell potential trading partners the services the company offers and the documents to use when communicating" does not teach or suggest using XML for messages or including an XML representation of an event in a message.

Thus, the rejection of claim 13 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks apply to claims 26 and 48 as well.

**Claim 34:**

Appellants have argued that, contrary to the Examiner's assertion, **Bass in view of Meltzer fails to teach or suggest a message in a data representation language including a data representation language representation of an event, where the event is a JAVA event**. The Examiner admits that Bass does not disclose the use of XML or JAVA events. The Examiner relies upon Meltzer, citing column 14, lines 25-32. Meltzer teaches a system for exchanging business related information using self-defining documents, such as XML based documents. The Examiner's cited passage describes

turning an XML document into a set of Java events. However, as noted above regarding claim 12, Meltzer teaches that listeners in a publish and subscribe architecture do not have to understand XML, but instead may be JAVA based and Meltzer specifically teaches the translation of XML documents into JAVA events to allow listeners to processing received documents (Meltzer, column 14, lines 41-54). Thus, rather than teaching or suggesting using data representation language representations of JAVA events, as the Examiner contends, Meltzer teaches the opposite, using Java events to represent a data representation language (e.g. XML) document.

Thus, the combination of Bass and Meltzer would not result in a publish and subscribe system in which messages including data representation language representations of JAVA events, as recited in Appellants' claims. Instead, the Examiner's proposed combination of Bass and Meltzer would result in a publish and subscribe system in which data representation language documents are translated into Java events for listeners or other recipients of the documents.

Furthermore, since, as noted above regarding the rejection of claim 12, Bass fails to teach the use of XML or JAVA events, as the Examiner admits, it would not make sense, nor would it be obvious, to modify the system of Bass to translate XML documents into JAVA events as taught by Meltzer. The Examiner's stated motivation, namely, so that a transaction process front end is able to operate in a publish and subscribe architecture that enables the addition of new listener programs without the knowledge of or impact on other listening programs in the system" is merely a reason why someone would use Meltzer's system and does not provide any reason to modify the teachings of Bass. As noted above, the Examiner is relying upon the teachings of Meltzer regarding translating XML documents to JAVA events. However, since Bass does not teach anything about either XML documents or JAVA events, there is no reason to modify Bass' system to include these teachings of Meltzer.

In the Examiner's Answer, the Examiner repeats the same response and arguments from the Examiner's response to Appellants' arguments regarding claim 12,



discussed above. Specifically, the Examiner completely ignores Appellants' arguments regarding Meltzer teaching the opposite of what the Examiner is relying up Meltzer to teach. The Examiner also ignores Appellants' arguments regarding a lack of a reason to modify Bass' system to include the teachings of Meltzer. Please refer to Appellants' discussion of claim 12 above regarding Examiner's remarks from the Examiner's Answer.

**Claim 35:**

Regarding claim 35, the Examiner's combination of **Bass in view of Meltzer fails to teach or suggest a service process configured to generate a message in a data representation language, wherein the message includes a data representation language representation of the event generated by the service process, wherein the data representation language is eXtensible Markup Language (XML)**. The Examiner admits that Bass fails to teach using XML and relies upon Meltzer, citing column 14, lines 25-32. As noted above regarding the rejection of claims 12 and 13, Meltzer teaches a system for exchanging business related information using self-defining documents, such as XML based documents. The Examiner's cited passage describes turning an XML document into a set of Java events. Thus, Meltzer does not teach or suggest sending messages in a XML, nor does Meltzer teach or suggest including an XML representation of an event in such a message. In fact, Meltzer teach away from including XML representations of events. Instead, as described above regarding claim 12, Meltzer teaches translating XML documents into JAVA events. Nowhere does Meltzer teach or suggest including an data representation language representation of an event in a message in a data representation language, where the data representation language is XML.

Since neither Bass nor Meltzer teach or suggest using XML for message or including an XML representation of an event in a message, the Examiner's combination of Bass and Meltzer also fails to teach or suggest sending messages in a XML, nor does Meltzer teach or suggest including an XML representation of an event in such a message.

Instead, the Examiner's combination of Bass and Meltzer would result in the publish and subscribe system taught by Bass that also includes translating XML documents into JAVA events, as taught by Meltzer. No combination of Bass and Meltzer would include messages in a data representation language or data representation language representations of events, where the data representation language is XML. Thus, the rejection of claim 35 is not supported by the prior art and removal thereof is respectfully requested.

In the Examiner's Answer, the Examiner repeats the same response and arguments from the Examiner's response to Appellants' arguments regarding claim 13, discussed above. Specifically, the Examiner has failed to response to Appellants' arguments regarding the fact that neither Bass nor Meltzer, whether considered singly or in combination, teach or suggest using XML for message or including an XML representation of an event in a message. Please refer to Appellants' discussion of claim 13 above regarding Examiner's remarks from the Examiner's Answer.

## **CONCLUSION**

For the foregoing reasons submitted in the Appeal Brief and this Reply Brief, it is submitted that the Examiner's rejections of claims 1-48 are erroneous. Reversal of the Examiner's decision is respectfully requested.

The Commissioner is authorized to charge any fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-65700/RCK.

Respectfully submitted,

/Robert C. Kowert/

Robert C. Kowert, Reg. #39,255  
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8850

Date: April 9, 2008